

# Documentação REST do projeto Saints

<b>RECURSO UTILIZADORES (/API/USERS)</b>	<b>2</b>
<b>RECURSO BARESDISCOTECAS (/API/BARESDISCOTECA )</b>	<b>6</b>
<b>RECURSO COMENTARIO (/API/COMENTARIO )</b>	<b>12</b>
<b>RECURSO CPERMANENTE (/API/CPERMANENTE)</b>	<b>16</b>
<b>RECURSO IMAGENS (/API/IMAGENS)</b>	<b>20</b>
<b>RECURSO LOGIN (/API/LOGIN)</b>	<b>24</b>
<b>RECURSO POST (/API/POST)</b>	<b>25</b>
<b>RECURSO TIPO (/API/TIPO)</b>	<b>29</b>

## Recurso utilizadores (/api/users)

### Obter todos os utilizadores

Devolve a lista de todos os utilizadores, como também toda a sua informação (Id, nome, senha, telemóvel, data de nascimento, atividade, posts e comentários).

/api/users (get)

Sucesso:

```
[
  {
    "id": 1,
    "usrName": "Rebeca",
    "usrSenha": "12345",
    "usrBdate": "2002-04-29",
    "usrTele": "969 596 640",
    "usrActive": false,
    "posts": [
      {
        "id": 2,
        "postUsrId": 1,
        "postTxt": "https://yt3.ggpht.com/a/AATXAJwuToYnwwkIvYe--
1RXXHRWWb6Jz-vAQ29qiQ=s900-c-k-c0xffffffff-no-rj-mo",
        "postVisivel": true,
        "postBd": 11
      }
    ],
    "comentarios": [],
    "cpermanentes": [
      {
        "id": 1,
        "compBdId": 2,
        "compTxt": "Espaço lindo",
        "compUsr": 1
      }
    ]
  }
]
```

...]

Erro:

404: "Not found"

Exemplo:

```
@GetMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)
public Iterable<User> getUsers() {
    logger.info("Sending all units");
    return userRepository.findAll();
}
```

### Inserir um novo utilizador

Inserir um utilizador sem comentários nem posts.

/api/users (post)

Dados:

```
{
  "usrName": "mario",
  "usrSenha": "wohoo",
  "usrBdate": "1997-03-04",
  "usrTele": "910600376",
  "usrActive": true,
  "posts": [],
  "comentarios": [],
  "cpermanentes": []
}
```

Sucesso:

```
{
  "id": 15,
  "usrName": "mario",
  "usrSenha": "wohoo",
  "usrBdate": "1997-03-04",
  "usrTele": "910600376",
  "usrActive": true,
  "posts": [],
  "comentarios": [],
  "cpermanentes": []
}
```

Erros:

400: Bad Request

500: Erro do servidor

Exemplo:

```
@PostMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)
public User addUser(@RequestBody User user) {
    logger.info("Saving user: {}", user.getUsrName());
    return userRepository.save(user);
}
```

### Eliminar um utilizador

Elimina um utilizador como toda a sua informação

/api/users (Delete)

Dados:

```
{
  {
    "id": 14,
    "usrName": "Martim",
    "usrSenha": "Martinhocas",
    "usrBdate": "1990-06-03",
    "usrTele": "879 000 595",
    "usrActive": false,
    "posts": [],
    "comentarios": [],
    "cpermanentes": []
  }
}
```

Sucesso:

Erros:

400: Bad Request

500: Erro do servidor

Exemplo:

```
@DeleteMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)
public void deleteUser(@RequestBody User user) {
    logger.info("Deleting user: {}", user.getUsrName());
    userRepository.delete(user);
}
```

### Atualizar um utilizador

Atualiza a informação de um utilizador

/api/users (patch)

Dados:

```
{
  {
    "id": 14,
    "usrName": "Martim",
    "usrSenha": "Martinhocas",
    "usrBdate": "1990-06-03",
    "usrTele": "879 000 595",
    "usrActive": false,
    "posts": [],
    "comentarios": [],
    "cpermanentes": []
  }
}
```

Sucesso:

Erros:

400: Bad Request

500: Erro do servidor

Exemplo:

```
@PatchMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)
public void updateUser(@RequestBody User user) {
    logger.info("Deleting user: {}", user.getUsrName());
    userRepository.save(user);
}
```

## Recurso BaresDiscotecas (/api/baresdiscoteca )

### Obter todos os bares e discotecas

Devolve a lista de todos os bares e discotecas, como também a sua informação (id,nome, localização, contacto, horário, tipo)

/api/baresdiscoteca (get)

Sucesso:

```
[  
  {  
    "id": 1,  
    "bdName": "Mome",  
    "bdLoc": "Edifício 78, Santos, R. da Cintura do Porto de Lisboa Armazém i, 1200-  
109 Lisboa ",  
    "bdHorario": "Todos os dias, menos terça - 23:59 às 06:00",  
    "bdContact": "211 510 734",  
    "bdTipo": 1  
  }  
...]
```

Erros:

404: "Not found"

Exemplo:

```
@GetMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)  
public Iterable<BaresDiscoteca> getBaresDiscotecas() {  
    logger.info("Retrieving all baresdiscotecas");  
    return baresdiscotecasRepository.findAll();  
}
```

## Inserir uma nova discoteca ou bar

Insere uma discoteca ou bar

/api/comentario (post)

Parâmetros:

**bdTipo**(obrigatório): bdTipo = 1, indica que é uma Discoteca, bdTipo = 2, indica que é um Bar

Dados:

```
{
  "bdName": "DANZA KUDURO",
  "bdLoc": "Edifício 80, Santos, R. da Cintura do Porto de Lisboa Armazém i, 1200-109 Lisboa ",
  "bdHorario": "Todos os dias, menos terça - 23:59 às 06:00",
  "bdContact": "8888 510 734",
  "bdTipo": 1
}
```

Sucesso (200):

```
{
  "id": 43,
  "bdName": "DANZA KUDURO",
  "bdLoc": "Edifício 80, Santos, R. da Cintura do Porto de Lisboa Armazém i, 1200-109 Lisboa ",
  "bdHorario": "Todos os dias, menos terça - 23:59 às 06:00",
  "bdContact": "8888 510 734",
  "bdTipo": 1
}
```

Erros:

500: Erro do servidor

400: Bad request

Exemplo:

```
@PostMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)
public BaresDiscoteca addBaresDiscotecas(@RequestBody BaresDiscoteca
baresDiscoteca) {

    logger.info("Saving baresdiscotecas: {}", baresDiscoteca.getBdName());
    return baresdiscotecasRepository.save(baresDiscoteca);
}
```

## Eliminar um bar ou discoteca

Elimina um bar ou discoteca como toda a sua informação

/api/users **(Delete)**

Dados:

```
{
  {
    "id": 14,
    "usrName": "Martim",
    "usrSenha": "Martinhocas",
    "usrBdate": "1990-06-03",
    "usrTele": "879 000 595",
    "usrActive": false,
    "posts": [],
    "comentarios": [],
    "cpermanentes": []
  }
}
```

Sucesso:

Erros:

400: Bad Request

500: Erro do servidor

Exemplo:

```
@DeleteMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)
public void deleteBaresDiscotecas(@RequestBody BaresDiscoteca
baresDiscoteca) {
    logger.info("Deleting baresdiscotecas: {}",
baresDiscoteca.getBdName());
    baresdiscotecasRepository.delete(baresDiscoteca);
}
```



### Atualizar um bar ou discoteca

Atualiza a informação de um bar ou discoteca

/api/users (patch)

Dados:

```
{
  "id": 43,
  "bdName": "DANZA KUDURO2",
  "bdLoc": "Edifício 80, Santos, R. da Cintura do Porto de Lisboa Armazém i, 1200-109 Lisboa ",
  "bdHorario": "Todos os dias, menos terça - 23:59 às 06:00",
  "bdContact": "8888 510 734",
  "bdTipo": null
}
```

Sucesso:

Erros:

400: Bad Request

500: Erro do servidor

Exemplo:

```
@PatchMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)
public void updateBaresDiscotecas(@RequestBody BaresDiscoteca baresDiscoteca)
{
    logger.info("Update baresdiscotecas: {}", baresDiscoteca.getBdName());
    baresdiscotecasRepository.save(baresDiscoteca);
}
```

**Obter a informação de um bar ou discoteca dado o nome**

Vai devolver a informação do bar ou discoteca (id, localização, contacto, horário)

/api/baresdiscoteca/name (**get**)

Parâmetros:

**Name** : Nome de uma discoteca ou bar já listados

Sucesso (200):

```
{
  "id": 11,
  "bdName": "Hype",
  "bdLoc": "Av. 24 de Julho 66, Lisboa 1200-869",
  "bdHorario": "Sexta-Sábado: 23:00-06:00",
  "bdContact": "https://www.instagram.com/hype.lisbon/",
  "bdTipo": 1
}
```

Erros:

500: Erro do servidor

Exemplo:

```
@GetMapping(path =("/{name}", produces = MediaType.APPLICATION_JSON_VALUE)
public BaresDiscoteca getBareDiscotecaByName(@PathVariable String name) {
    name = URLDecoder.decode(name, StandardCharsets.UTF_8);
    logger.info("Retrieving object from name: {}", name);
    logger.info("Retrieved: {}",
baresdiscotecasRepository.findBaresDiscotecaByBdName(name).getBdName());
    return baresdiscotecasRepository.findBaresDiscotecaByBdName(name);
}
```

**Obter o tipo de um bar ou discoteca**

Vai devolver o tipo e a informação do bar ou discoteca.

/api/baresdiscotecas/tipo/{tpName} (get)

Parâmetros:

tpName (obrigatório): Bar ou Discoteca (obrigatório)

Sucesso (200):

```
{
  "id": 1,
  "bdName": "Mome",
  "bdLoc": "Edifício 78, Santos, R. da Cintura do Porto de Lisboa Armazém i, 1200-109 Lisboa ",
  "bdHorario": "Todos os dias, menos terça - 23:59 às 06:00",
  "bdContact": "211 510 734",
  "bdTipo": 1
}
```

Erros:

500: Erro do servidor

Exemplo:

```
@GetMapping(path = "/tipo/{tpName}", produces = MediaType.APPLICATION_JSON_VALUE)
public Iterable<BaresDiscoteca> getBareDiscotecaByTipo(@PathVariable String tpName) {
    Tipo tipo = tipoRepository.findTipoByTpNome(tpName);
    return
baresdiscotecasRepository.findBaresDiscotecasByBdTipoId(tipo.getId());
}
```

## Recurso comentario (/api/comentario )

### Obter todos os comentários

Devolve a lista de todos os comentários, como também toda a sua informação (Id, usrid, texto, visibilidade, postid).

/api/comentario **get**)

Sucesso:

```
[  
  {  
    "id": 1,  
    "usr_id": 3,  
    "post_id": 4,  
    "txt": "Está vazio. NÃO VENHAM",  
    "visivel": true  
  }  
...]
```

Erro:

404: "Not found"

Exemplo:

```
@GetMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)  
public Iterable<Comentario> getComentarios() {  
    logger.info("Retrieving all comments");  
    return comentarioRepository.findAll();  
}
```

### Inserir um novo comentário

Vai devolver a informação do álbum, do artista e das músicas do álbum

/api/comentario (post)

```
{
  "usr_id": 3,
  "post_id": 4,
  "txt": "NÃO VENHAM",
  "visivel": true
}
```

Sucesso (200):

```
{
  "id": 13,
  "usr_id": 3,
  "post_id": 4,
  "txt": "NÃO VENHAM",
  "visivel": true
}
```

Erros:

500: Erro do servidor

400: Bad request

Exemplo:

```
@PostMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)
public Comentario addComentario(@RequestBody Comentario comentario) {

    logger.info("Saving comment: {}", comentario.getTxt());
    return comentarioRepository.save(comentario);
}
```

## Eliminar um comentário

Elimina um comentário

/api/comentario (**delete**)

```
{
  "id": 1,
  "usr_id": 3,
  "post_id": 4,
  "txt": "Está mau. NÃO VENHAM",
  "visivel": true
}
```

**Sucesso (200):**

Erros:

500: Erro do servidor

400: Bad request

Exemplo:

```
@DeleteMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)
public void deleteComentario(@RequestBody Comentario comentario) {
    logger.info("Deleting comment: {}", comentario.getTxt());
    comentarioRepository.delete(comentario);
}
```

### Atualizar um comentário

Atualiza a informação de um comentário

/api/comentario (patch)

Dados:

```
{
  "id": 1,
  "usr_id": 3,
  "post_id": 4,
  "txt": "Está vazio.",
  "visivel": true
}
```

**Sucesso (200):**

Erros:

400: Bad Request

500: Erro do servidor

Exemplo:

```
@PatchMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)
public void updateComentario(@RequestBody Comentario comentario) {
    logger.info("Update comment: {}", comentario.getTxt());
    comentarioRepository.save(comentario);
}
```

## Recurso cPermanente (/api/cpermanente)

### Obter todos os comentários permanentes

Devolve a lista de todos os comentários permanentes, como também toda a sua informação (Id, compBdId, texto, compUsr).

/api/cpermanente (get)

Sucesso:

```
{
  "id": 1,
  "compBdId": 2,
  "compTxt": "Espaço lindo",
  "compUsr": 1
}
...]
```

Erro:

404: "Not found"

Exemplo:

```
@GetMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)
public Iterable<Cpermanente> getCPermanente() {
    logger.info("Retrieving all comments");
    return cPermanenteRepository.findAll();
}
```



## Inserir um novo comentário permanente

Insere um novo comentário

/api/cpermanente **(post)**

```
{  
  
    "compBdId": 23,  
    "compTxt": "as vezes e mau",  
    "compUsr": 7  
}
```

Sucesso (200):

```
{  
  
    "id": 14,  
    "compBdId": 23,  
    "compTxt": "as vezes e mau",  
    "compUsr": null  
}
```

Erros:

500: Erro do servidor

400: Bad request

Exemplo:

```
@PostMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)  
public Cpermanente addCPermanente(@RequestBody Cpermanente cPermanente) {  
  
    logger.info("Saving comment: {}", cPermanente.getCompTxt());  
    return cPermanenteRepository.save(cPermanente);  
}
```

## Eliminar um comentário permanente

Elimina um comentário permanente

/api/cpermanente (delete)

```
{
  "id": 1,
  "usr_id": 3,
  "post_id": 4,
  "txt": "Está mau. NÃO VENHAM",
  "visivel": true
}
```

**Sucesso (200):**

Erros:

500: Erro do servidor

400: Bad request

Exemplo:

```
@DeleteMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)
public void deleteCPermanente(@RequestBody Cpermanente cPermanente) {
    logger.info("Deleting comment: {}", cPermanente.getCompTxt());
    cPermanenteRepository.delete(cPermanente);
}
```

### Atualizar um comentário permanente

Atualiza a informação de um comentário permanente

/api/cpermanente (patch)

Dados:

```
{
  "id": 1,
  "imgCeoId": 2,
  "imgUrl": "https://clubbable.blob.core.windows.net/medias/MOME-
?timestamp=636777100181475061"
}
```

**Sucesso (200):**

Erros:

400: Bad Request

500: Erro do servidor

Exemplo:

```
@PatchMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)
public void updateCPermanente(@RequestBody Cpermanente cPermanente) {
    logger.info("Update comment: {}", cPermanente.getCompTxt());
    cPermanenteRepository.save(cPermanente);
}
```

## Recurso Imagens (/api/Imagens)

### Obter todas as imagens

Devolve a lista de todas as imagens (url) como o seu id e o id do CEO.

/api/Imagens (get)

Sucesso (200):

```
[  
  {  
    "id": 1,  
    "imgCeoId": 2,  
    "imgUrl": "https://media-cdn.tripadvisor.com/media/photo-s/12/d4/00/55/piso-  
0-club.jpg"  
  }  
...]
```

Erro:

404: "Not found"

Exemplo:

```
@GetMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)  
public Iterable<Imagens> getImagens() {  
    logger.info("Retrieving all Imagens");  
    return imagensRepository.findAll();  
}
```

<b>Inserir uma nova Imagem</b> Insere uma nova Imagem
/api/Imagens (post)
Parâmetros: <b>imgBdId</b> (obrigatório): id da discoteca ou bar onde vai inserir a imagem
Dados: <pre> {     "id": 4,     "imgCeoId": 1,     "imgUrl": "https://clubbable.blob.core.windows.net/medias/MOME- ?timestamp=636777100181475061",     "imgBdId": 1 } </pre>
Sucesso (200): <pre> {     "id": 4,     "imgCeoId": 1,     "imgUrl": "https://clubbable.blob.core.windows.net/medias/MOME- ?timestamp=636777100181475061" } </pre>
Erros: 500: Erro do servidor 400: Bad request
Exemplo: <pre> @PostMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE) public Imagens addImagens(@RequestBody Imagens imagens) {      logger.info("Saving imagens: {}", imagens.getImgUrl());     return imagensRepository.save(imagens); } </pre>

## Eliminar uma imagem

Elimina uma imagem

/api/imagens (delete)

```
{
  "id": 4,
  "imgCeoId": 1,
  "imgUrl": "https://clubbable.blob.core.windows.net/medias/MOME-
?timestamp=636777100181475061",
  "imgBdId": 1
}
```

**Sucesso (200):**

Erros:

500: Erro do servidor

400: Bad request

Exemplo:

```
@DeleteMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)
public void deleteimagens(@RequestBody Imagens imagens) {
    logger.info("Deleting imagens: {}", imagens.getImgUrl());
    imagensRepository.delete(imagens);
}
```

### Atualizar uma imagem

Atualiza a informação de uma imagem

/api/imagens (patch)

Dados:

```
{
  "id": 1,
  "imgCeoId": 2,
  "imgUrl": "https://media-cdn.tripadvisor.com/media/photo-s/12/d4/00/55/piso-0-club.jpg",
  "imgBdId": 1
}
```

**Sucesso (200):**

Erros:

400: Bad Request

500: Erro do servidor

Exemplo:

```
@PatchMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)
public void updateimagens(@RequestBody Imagens imagens) {
    logger.info("Update imagens: {}", imagens.getImgUrl());
    imagensRepository.save(imagens);
}
```

## Recurso Login (/api/login)

### Fazer login

Recebe o nome e password do utilizador caso esse exista recebe a mensagem "ok", caso contrário aparecerá a mensagem "Invalid credentials",

/api/login (**post**)

Dados:

```
{ "username": "Rebeca",  
  "password": "12345"  
}
```

Sucesso (200):

OK

Caso as credencias estejam mal ou não existam:

Invalid credentials

Erros:

500: Erro do servidor

400: Bad request

```
@PostMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)  
@ResponseBody  
public String login(@RequestBody LoginUser loginUser) {  
    User user = userRepository.findUserByUsrName(loginUser.getUsername());  
    if(user == null) {  
        return "User is null";  
    }logger.info(loginUser.getUsername());  
    logger.info(loginUser.getPassword());  
    logger.info(user.getUsrSenha());  
    if(user.getUsrSenha().equals(loginUser.getPassword())) {  
        return "OK";  
    }  
    return "Invalid credentials";  
}
```



## Recurso Post (/api/post)

### Obter todos os Post

Devolve a lista de todos os posts tal como a sua informação (id,etc)

/api/post (**get**)

Sucesso (200):

```
[  
  {  
    "id": 1,  
    "postUsrId": 2,  
    "postTxt": "https://lifecooler.com/files/registos/imagens/461421/400794.jpg",  
    "postVisivel": true,  
    "postBd": 1  
  }  
...]
```

Erro:

404: "Not found"

Exemplo:

```
@GetMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)  
public Iterable<Post> getPosts() {  
    logger.info("Retrieving all posts");  
    return postRepository.findAll();  
}
```

## Inserir um novo post

Insere um novo post

/api/cpermanente (post)

Dados:

```
{
  "id": 5,
  "postTxt": "https://lifecooler.com/files/registos/imagens/461421/400794.jpg",
  "postVisivel": true,
  "postBd": 1,
  "postUsrId": 2
}
```

Sucesso (200):

```
{
  "id": 5,
  "postTxt": "https://lifecooler.com/files/registos/imagens/461421/400794.jpg",
  "postVisivel": true,
  "postBd": 1,
  "postUsrId": 2
}
```

Erros:

500: Erro do servidor

400: Bad request

Exemplo:

```
@@PostMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)
public Post addPost(@RequestBody Post post) {

    logger.info("Saving post: {}", post.getPostTxt());
    return postRepository.save(post);
}
```

## Eliminar um post

Elimina um post

/api/post (**delete**)

```
{
  "id": 4,
  "imgCeoId": 1,
  "imgUrl": "https://clubbable.blob.core.windows.net/medias/MOME-
?timestamp=636777100181475061",
  "imgBdId": 1
}
```

**Sucesso (200):**

Erros:

500: Erro do servidor

400: Bad request

Exemplo:

```
@DeleteMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)
public void deletePost(@RequestBody Post post) {
    logger.info("Deleting post: {}", post.getPostTxt());
    postRepository.delete(post);
}
```

### Atualizar um post

Atualiza a informação de um post

/api/post (**patch**)

Dados:

```
{
  "id": 1,
  "postTxt": "https://lifecooler.com/files/registos/imagens/461421/400794.jpg",
  "postVisivel": true,
  "postBd": 1,
  "postUsrId": 2
}
```

**Sucesso (200):**

Erros:

400: Bad Request

500: Erro do servidor

Exemplo:

```
public void updatePost(@RequestBody Post post) {
    logger.info("Update post: {}", post.getPostTxt());
    postRepository.save(post);
}

@PatchMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)
public void updateUser(@RequestBody Post post) {
    logger.info("Update post: {}", post.getPostTxt());
    postRepository.save(post);
}
```

## Recurso Tipo (/api/tipo)

### Obter o tipo

Vai devolver os tipos (o id, como também o nome)

/api/tipo (get)

Sucesso (200):

```
[
  {
    "id": 1,
    "tpNome": "Discoteca"
  },
  {
    "id": 2,
    "tpNome": "Bar"
  }
]
```

Erros:

500: Erro do servidor

Exemplo:

```
@GetMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)
public Iterable<Tipo> getTipos() {
    logger.info("Retrieving all tipos");
    return tipoRepository.findAll();
}
```

### Obter o tipo dado o id

Vai devolver o tipo a partir do id

/api/tipo/{id} (get)

Parâmetros:

Id (obrigatório): 1 = Discoteca, 2 = Bar

Sucesso (200):

```
[
  {
    "id": 1,
    "tpNome": "Discoteca"
  }
]
```

Erros:

500: Erro do servidor

Exemplo:

```
@GetMapping(path =("/{id}", produces = MediaType.APPLICATION_JSON_VALUE)
public Tipo getTipoById(@PathVariable Integer id) {
    logger.info("Retrieving tipo by id: {}", id);
    return tipoRepository.findTipoById(id);
}
```

## Eliminar um tipo

Elimina um tipo

/api/tipo (delete)

```
{
  "id": 3,
  "tpNome": "Twerk"
}
```

**Sucesso (200):**

Erros:

500: Erro do servidor

400: Bad request

Exemplo:

```
@DeleteMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)
public void deleteTipo(@RequestBody Tipo tipo) {
    logger.info("Deleting tipo: {}", tipo.getTpNome());
    tipoRepository.delete(tipo);
}
```

### Inserir um novo tipo

Insere um novo id como também o nome do tipo.

/api/tipo (post)

Parâmetros:

Id (obrigatório): 1 = Discoteca, 2 = Bar

Sucesso (200):

```
{
  "id": 3,
  "tpNome": "Twerk"
}
```

Erros:

400: Bad request

500: error server

Exemplo:

```
@PostMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)
public Tipo addTipo(@RequestBody Tipo tipo) {
    logger.info("Saving tipo: {}", tipo.getTpNome());
    return tipoRepository.save(tipo);
}
```



### Atualizar um tipo

Atualiza a informação de um tipo

/api/tipo(patch)

Dados:

```
{
    "id": 3,
    "tpNome": "twerk3"
}
```

**Sucesso (200):**

Erros:

400: Bad Request

500: Erro do servidor

Exemplo:

```
@PatchMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)
public void updateTipo(@RequestBody Tipo tipo) {
    logger.info("Update tipo: {}", tipo.getTpNome());
    tipoRepository.save(tipo);
}
```